웨어러블 기반 라이프로그 데이터를 활용한 치매 예측을 위한 다양한 딥러닝 접근 방식 탐색

2017741007 김진기

JinGi Kim

Abstract: 본 연구는 웨어러블 기반 라이프로그 데이터를 활용하여 치매 예측 성능을 향상시키기 위해 다양한 접근 방식을 탐색하였습니다. 데이터 처리 단계에서는 라이프로그 데이터를 그래프로 변환하고, ResNet+VAE와 VAE를 사용하여 잠재 벡터를 추출하여 변수로 추가하는 방식과 PCA 방식을 비교 분석하였습니다. 또한, 직접 구현한 ResNet, CNN+LSTM, LSTM 앙상블과 같은 예측 모델을 활용하여 성능을 평가하고 비교하였습니다. 실험 결과, LSTM 앙상블과 ResNet 모델이 전반적으로 강력한 예측 성능을 보였으며, CNN+LSTM 모델은 상대적으로 약한 결과를 나타냈습니다. 또한, 변수로 추가한 접근 방식이 재현 성능을 더욱 향상시켰음을 확인하였습니다. 이러한 연구 결과는 웨어러블 기반 라이프로그 데이터를 활용하여 치매 예측에 LSTM 앙상블과 ResNet과 같은 딥러닝 모델이 유용하다는 것을 보여줍니다. 이를 토대로 더 정확하고 실용적인 치매 예측 모델의 개발을 위한 향후 연구에 기여할 것입니다.

1. 서 론

2020년 기준, 65세 이상의 노인 인구 중에서 10% 이상이 치매를 겪고 있으며, 경도 인지 장애와 같은 치매의 초기 단계에 해당하는 질환이 60세 기준으로 20% 이상인 것으로 알려져 있습니다.[10] 치매는 국가적으로도 큰 피해를 주는데 뿐만 아니라 가정 내에서도 큰 영향을 미치기 때문에 이 문제를 꼭 해결해야 합니다. 그러나 현재 치매를 측정하기 위한 방법은 매우 어렵고 비용도 많이 들어가기 때문에 현실적으로 어렵습니다.

따라서, 웨어러블 기반 데이터 수집을 통해 별도의 시간을 투자하지 않고도 수집된 데이터를 기반으로 치매와 경도 인지 장애를 조기에 예측할 수 있는 치매 예측 모델을 개발하고자 합니다. 이를 위해 라이프로그 데이터를 전처리하고 변수를 추가하여 모델에 학습시킬 계획입니다. 학습된 모델들을 비교 분석하여 최상의 결과를 도출하고, 이를 토대로 치매 예측 모델의 개발과 활용에 기여하고자 합니다.

요약하자면, 이 연구는 치매 예측을 위해 웨어러블 데이터를 활용하고자 하며, 데이터 전처리와 변수 추가를 통해 모델을 학습시키고 결과를 분석하는 과정을 진행할 예정입니다. 이를 통해 조기 치매 예측 및 대응에 기여하고자 합니다.

2. 본 론

2.1 데이터셋

치매 예방을 위한 라이프로그 치매 분류
https://aihub.or.kr/problem_contest/nipa-learning-platform/6

2.2 데이터 처리

2.2.1 Data drop

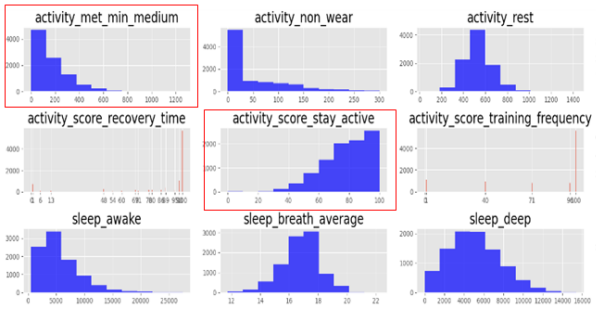
```
tab_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9327 entries, 0 to 9326
Data columns (total 66 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   EMAIL                   9327 non-null   object
 1   summary_date            9327 non-null   datetime64[ns]
 2   activity_average_met    9327 non-null   float64
 3   activity_cal_active     9327 non-null   int64
 4   activity_cal_total      9327 non-null   int64
 5   activity_class_5min     9327 non-null   object
```

[Fig. 1] Data information

[Fig 1]과 같이 데이터의 정보를 불러와 데이터의 내용 확인 및 학습 시 필요 없는 데이터와 값을 바꿔줘야 하는 데이터를 선택해 값을 데이터를 수정했습니다.

2.2.2 Data transform,



[Fig. 2] Data distribution

[Fig 2]을 통해 좌측 편향된 데이터는 Log Transform[2]

을 우측 편향된 데이터는 Quantile Transform[3]을 사용해 분포를 보다 정규분포에 가깝게 만들어 모델이 보다 데이터의 분포를 잘 파악하게 했습니다.

2.2.3 Data scaling

```
# Scaling
# processed -> processed1
standardScaler = RobustScaler()
scaled_data = processed_data.iloc[:, :48].copy()
scaled_data = pd.DataFrame(standardScaler.fit_transform(scaled_data), columns = scaled_data.columns)
processed_data.iloc[:, :48]=scaled_data

processed_data
processed_data1 = processed_data.copy()
```

	activity_average_met	activity_cal_active	activity_cal_total	activity_daily_movement	activity_high
0	1.285714	0.710130	1.086168	0.734232	-0.731104
1	-0.142857	-0.245737	-0.036281	-0.243021	-0.731104
2	0.142857	-0.044132	0.179138	-0.088055	-0.731104

[Fig. 3] Data scaling

[Fig 3]과 같이 RobustScaler[4]을 이용해 이상치에 강하며 데이터의 특성을 평균이 0이고 표준 편차가 1인 분포로 변환하는 작업을 했습니다.

2.3 PCA[1]를 이용한 데이터 추가

2.3.1 차원축소를 이용한 데이터 추가

```
pca = PCA(n_components=4)
processed_data_X = pca.fit_transform(processed_data_X)
processed_data_X = pd.DataFrame(processed_data_X)

pca = PCA(n_components=4)
processed_data_X = pca.fit_transform(processed_data_X)
processed_data_X = pd.DataFrame(processed_data_X)

pca_data = pd.concat([processed_data_X, processed_data_y], axis=1)
pca_data
```

	0	1	2	3	DIAG_NM
0	-0.913480	-0.930877	-2.478338	1.561210	CN
1	0.995512	-0.730126	-1.441353	-0.155199	CN
2	0.654839	-0.819732	-1.475013	-0.084798	CN
3	4.987305	1.198441	0.142355	-5.016961	CN
4	1.290475	-0.920937	-1.089832	-0.579167	CN

[Fig. 4] PCA

[Fig 4]와 같이 2.2.3의 과정만을 거친 데이터셋을 4개의 차원으로 축소하는 과정을 거쳤습니다.

2.3.2 RandomForest[5]와 차원축소를 이용한 데이터 추가

[Fig. 5] Random + PCA

[Fig 5]와 같이 2.2.3의 과정을 거친 데이터셋을 Randomforest[5]를 통해 importance feature을 4개의 그룹으로 구분한 뒤 그룹을 하나의 차원으로 합쳐서 총 4개의 변수로 생성하였다.

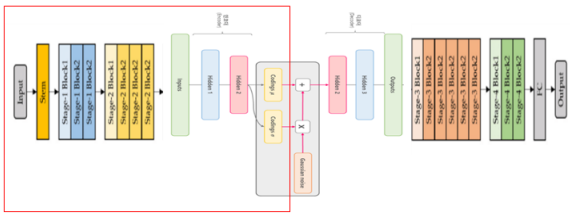
2.4 그래프 이미지를 이용한 데이터 추가

2.4.1 라이프로그 시그널 데이터 그래프로 변형

[Fig. 6] Image Generate

분 단위로 측정된 로그데이터를 [Fig 6] 과정을 통해 데이터를 그래프로 만든 뒤 저장시켰습니다.

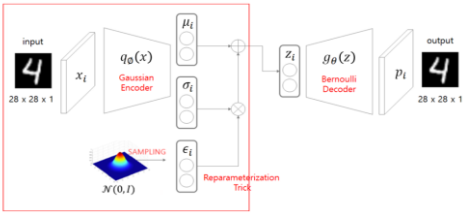
2.4.2 ResNet[6] + VAE[7]를 이용한 추가



[Fig. 7] ResNet[6] + VAE[7]

[Fig 7] 형태의 Resnet + VAE모델을 구현한 뒤 2.4.1의 과정을 통해 얻은 이미지 데이터를 해당 모델에 학습시킨 뒤 재구성 과정에 필요한 Latent vector만을 사용하기 위해 Encoder 과정만을 거치고 나온 결과를 하나의 차원으로 줄인 뒤 기존의 전 처리 과정을 거친 데이터셋에 추가하는 과정을 거쳤습니다.

2.4.3 VAE[7]를 이용한 추가



[Fig. 8] VAE[7] model

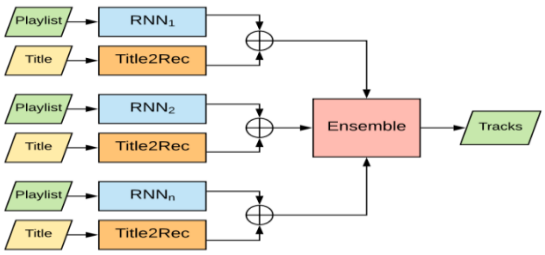
[Fig 7] 형태의 VAE모델을 구현한 뒤 2.4.1의 과정을 통해 얻은 이미지 데이터를 VAE에 넣어 학습시킨 뒤 재구성 과정에 필요한 Latent vector만을 사용하기 위해 Encoder 과정만을 거치고 나온 결과를 하나의 차원으로 줄인 뒤 기존의 전 처리 과정을 거친 데이터셋에 추가하는 과정을 거쳤습니다.

2.4.3 기본 데이터셋

[2.2.3]의 과정만을 거친 데이터 셋입니다.

2.5 모델 구성

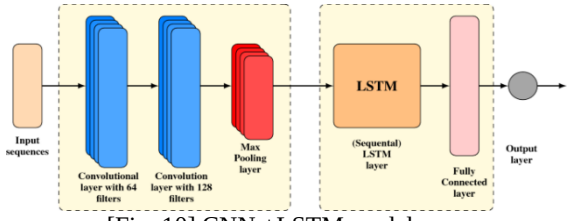
2.5.1 LSTM[8] 앙상블 모델 구현



[Fig. 9] LSTM Ensemble model

[Fig 9]와 같이 Keras를 이용해 Model을 구현했다.

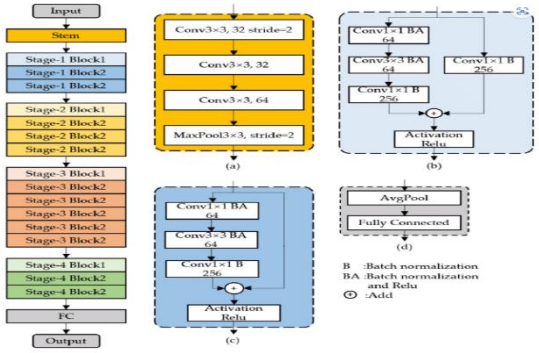
2.5.2 CNN[9] + LSTM[8] 모델 구현



[Fig. 10] CNN +LSTM model

[Fig 10]와 같이 Keras를 이용해 Model을 구현했다.

2.5.3 ResNet[6] 모델 구현



[Fig. 11] ResNet[6] model

[Fig 11]와 같이 Keras를 이용해 Model을 구현했다.

3. 실험 결과

3.1.1 Latent + PCA 차원축소

	LSTM Ensemble	CNN + LSTM	ResNet
precesion	0.72	0.64	**0.74**
recall	0.89	**1.0**	0.84
f1-score	**0.80**	0.78	0.78

[Table 1] 표 1

3.1.2 Latent + Randomforest와 PCA 차원축소

	LSTM Ensemble	CNN + LSTM	ResNet
precesion	**0.77**	0.64	0.71
recall	0.86	**1.0**	0.91
f1-score	**0.81**	0.78	0.80

[Table 2] 표 2

3.1.3 Latent + Processed

	LSTM Ensemble	CNN + LSTM	ResNet
precesion	0.78	0.64	**0.81**
recall	0.79	**1.0**	0.79
f1-score	**0.79**	0.78	0.80

[Table 3] 표 3

3.1.4 Processed Data

	LSTM Ensemble	CNN + LSTM	ResNet
precesion	0.78	0.64	**0.85**
recall	0.80	**1.0**	0.82
f1-score	0.79	0.78	**0.84**

[Table 4] 표 4

Recall은 치매 환자 중 진짜 치매로 예측될 확률 precision은 치매 진단 환자 중 진짜 치매일 확률을 뜻하며 대체적으로 Resnet과 Lstm Ensemble 모델이 높은 성능을 보여줌을 알 수 있으며, Latent 변수와 차원축소 변수 합성 시 Recall이 높은 성능을 보임을 알 수 있다.

4. 결 론

본 연구에서는 웨어러블 기반 라이프로그 데이터를 활용한 치매 예측을 위해 다양한 딥러닝 접근 방식을 탐색하였습니다. 연구 결과로써 다음과 같은 결론을 도출할 수 있습니다.

첫째, 웨어러블 기반 라이프로그 데이터를 활용하여 치매 예측을 수행하는 것은 유효하며, 딥러닝 모델을 활용할 때 높은 예측 성능을 얻을 수 있었습니다. 특히, LSTM 앙상블과 ResNet 모델은 뛰어난 성능을 보여주었습니다.

둘째, 잠재 벡터를 추가한 ResNet+VAE와 PCA 방식의

변수 추가는 예측 모델의 성능 향상에 기여하였습니다. 특히, 재현율 측면에서 잠재 변수와 차원 축소 변수의 합성이 높은 성능을 보였습니다.

이러한 연구 결과는 웨어러블 기반 라이프로그 데이터를 활용한 치매 예측에 딥러닝 접근 방식이 유용하다는 것을 보여줍니다. 또한, 치매 예측 모델의 개발과 관련하여 변수 추가와 차원 축소 기법의 활용이 성능 향상에 기여할 수 있다는 점을 확인하였습니다.

References

[1] Harold Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," Journal of Educational Psychology, Vol. 24, No. 7, pp. 498-520, 1933.

[2] Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. Journal of the Royal Statistical Society: Series B (Methodological), 26(2), 211-252.

[3] R.J. Hyndman and Y. Fan, "Sample Quantiles in Statistical Packages," The American Statistician, Vol. 50, No. 4, pp. 361-365, 1996.

[4] Rousseeuw, P.J., Croux, C. (1993). Alternatives to the Median Absolute Deviation. Journal of the American Statistical Association, Vol. 88, pp. 1273-1283.

[5] Leo Breiman, "Random Forests," Machine Learning, Vol. 45, No. 1, pp. 5-32, 2001.\

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. "Deep Residual Learning for Image Recognition", Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[7] Diederik P Kingma, Max Welling. "Auto-Encoding Variational Bayes", International Conference on Learning Representations, 2014.

[8] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[9] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[10]http://www.dailypharm.com/Users/News/NewsView.html?ID=296528